

SIMULADOR DE SEÑALES R.A.D.A.R

PARA VISUALIZACION EN UN P.P.I

A. AGUASCA SOLÉ* - A. ELIAS FUSTÉ* - A. BROQUETAS IBARS*

ABSTRACT:

This paper shows the transformation of a non-coherent X-band RADAR in a student training RADAR station. Using the RADAR P.P.I and a personal computer to control it, Rayleigh targets and clutter statistics are simulated and introduced in base band. Raw video signals and several post-processing data can be represented.

INTRODUCCION:

Se presenta el diseño de un simulador de Receptor RADAR a partir de un ordenador en configuración PC y la Unidad de Presentación Visual de un RADAR de banda-X a 9.35GHz con su F.I a 40MHz. La idea original es que mediante el ordenador se genere una señal digital que simule la detectada por envolvente de un RADAR monostático, incoherente, convertirla a analógica mediante un conversor D/A para luego ser inyectada por una entrada en banda base que se posee en la U.P.V y que será la visualizada en el P.P.I. Hasta aquí la función del conjunto PC-U.P.V sería la del sistema original del RADAR HALCON 948 conectado con su sistema radiante, pero para extenderse un poco dentro del mundo del procesado de datos RADAR, y ver sus posibilidades, el ordenador asume también la función de procesador de datos, como son:

- 1.- Detector por umbral fijo.
- 2.- Extractor.
- 3.- Detector por umbral adaptativo (C.F.A.R)

REALIZACIONES HARDWARE:

Vista la idea original del simulador, se debe ahora considerar las posibilidades del PC como emulador del sistema de recepción de un RADAR. Lo primero que surge como limitación es la velocidad de salida de los datos por un port paralelo de 8 bits como el incluido en la placa 8255 I/O CARD. Hecha una prueba mediante un generador de onda cuadrada implementado por software en lenguaje ASSEMBLER, se vió que la frecuencia máxima generable era de 90.000KHz, equivalente a tener limitada la extracción de datos en paralelo a 180.000 muestras/s.

Puesto que se trata de un ordenador secuencial, con un reloj de 8MHz, se ve rápidamente la imposibilidad de generar la señal detectada RADAR simultáneamente con la extracción por el port de salida, y por otra parte tampoco se puede generar en los posibles intervalos muertos entre extracciones ya que el lapso temporal entre la generación de dos datos RADAR es tal que sobrepasa el utilizable, por tanto solo resta la solución de poseer un banco de datos que corresponda a distintas pantallas, que serán generadas con anterioridad a su posible procesado o bien su extracción hacia el P.P.I., dichos bancos de datos serán la información que simula la señal detectada por envolvente agrupada en pantallas.

Así pues, hasta ahora tenemos limitada la velocidad de extracción de datos a 180.000 muestras/s, así como la necesidad de poseer una serie de bancos de datos que correspondan a una serie de "scans" del espacio.

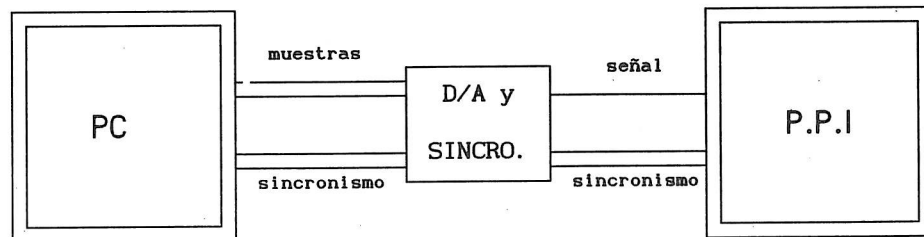
El alcance máximo del RADAR a simular es de 48MN, con posibilidad de

* Grupo A.M.R Departamento de TEORIA DE LA SEÑAL Y COMUNICACIONES U.P.C

escalado intermedio. Si se intentase construir un simulador de la recepción a un alcance de 48MN, puesto que la duración del impulso emitido es de $1\mu s$, la célula de resolución en distancia abarca 150m, lo que equivaldría a extraer en pantalla en un barrido radial de la misma de $580\mu s$ un total de 590 muestras, esto es, un volcado de datos hacia el conversor D/A a una frecuencia de aproximadamente 1000.000 muestras/s, totalmente fuera del alcance del PC. Pero por contra, si suponemos la duración de un barrido radial de todavía $580\mu s$ pero la escala a simular de 6MN tenemos que el número de muestras por barrido es de 100, que equivale a necesitar una velocidad de volcado de datos sobre los 172.000 muestras/s, que entra dentro de nuestras posibilidades. Así pues, la solución consiste en suponer un alcance de 6MN trabajando con la presentación práctica de la escala de 48 MN, equivalente a un efecto Zoom.

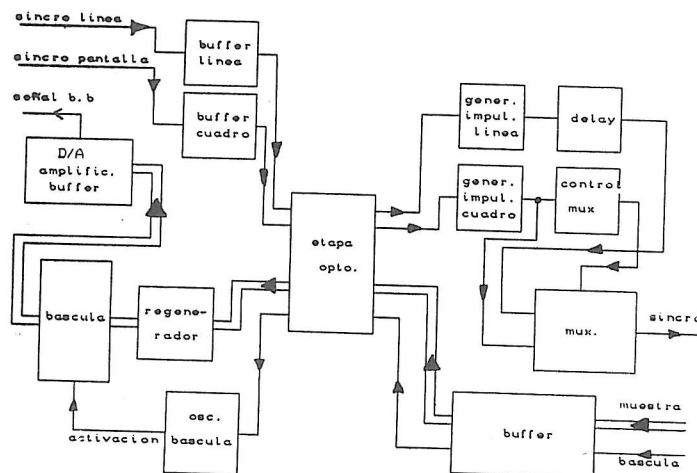
Pero todavía surge otra limitación, esta vez del número de muestras necesario para cubrir todo el espacio. Para conseguir la extracción de manera sincronizada, con duraciones idénticas de cada muestra, es necesario que estén almacenados en memoria RAM en un espacio inferior de 64KBytes, el motivo es bien sencillo, el microprocesador del PC, un INTEL 8088, direcciona una posición de memoria por segmentación, esto es, la dirección real de un byte de memoria es apuntada mediante dos registros, el registro segmento y el registro desplazamiento, el registro segmento apunta a un espacio de 64 Kbytes y el de desplazamiento a un elemento dentro de dicho espacio. Si se tuviese que trabajar sobre un segmento superior a los 64KBytes implicaría modificar el registro segmento entre extracciones, cuestión imposible puesto que podría generarse una desincronización en el volcado de las muestras. Así pues, tenemos limitado el número de muestras que constituyen un scan del espacio. Puesto que la resolución angular que posee el P.P.I es de $0,18^\circ$, y queremos simular un sistema cuyo ancho de haz es de $0,72^\circ$ (la mitad del sistema real), se recurre a la técnica de presentar durante cuatro barridos las mismas muestras, con ello se consigue presentar todo el espacio con un número de muestras de aproximadamente 50.000, suficientes para introducirlas en único segmento de memoria.

El esquema del sistema es:



La etapa de interface consta de dos partes, la unidad de conexión con el ordenador, encargado de darle los sincronismos necesarios, como son inicio de pantalla y de línea, así como una etapa aisladora mediante dispositivos optoelectrónicos por la cual pasan los sincronismos y las muestras en paralelo. La otra etapa se encarga de recoger los sincronismos generados por el P.P.I, transmitirlos hacia la primera etapa, y recoger las muestras entregadas por el PC, regenerarlas debido al tránsito de las mismas por la etapa optoacopladora, convertirlas mediante un D/A a señal analógica y un buffer que la inyecta a la entrada en banda base. Cabe destacar también una serie de básculas activadas por un oscilador controlado por una línea de sincronismo, ello es para evitar posibles ruidos de "glitch" a la salida del D/A debido a los diferentes tiempos de conmutación alto->bajo bajo->alto de los bits transmitidos a través de los optoacopladores. Para generarse los sincronismos de línea y de pantalla se

recurre a la técnica de conocer el instante de retrazado de la pantalla y, mediante un temporizador, generar el impulso de sincronismo de línea en el instante oportuno.



DETALLE DE LA ETAPA DE INTERFACE ENTRE P.P.I Y PC

REALIZACIONES SOFTWARE:

Puesto que se van a simular señales detectadas por envolvente a Frecuencia Intermedia, se hace necesario tener un conocimiento, aunque sencillo, de las características de dicha señal. En un principio suponemos el espacio libre de blancos con un fondo de CLUTTER que a 9,3GHz la potencia recibida tiene una estadística gaussiana. Si ahora se detecta por envolvente, una vez rebajada la información a F.I, la estadística de la potencia de la envolvente es de tipo RAYLEIGH con media igual a la desviación típica = σ_1 :

$$p_n(n) = \frac{n^2}{\sigma_1^2} e^{-\frac{n^2}{\sigma_1^2}} \quad n > 0 ;$$

Si una célula de resolución posee un blanco, que caracterizaremos por devolver un eco cuya potencia también se rige por una estadística gaussiana, la señal que se detectará, al igual que en el caso de CLUTTER, se registrará por una estadística tipo RAYLEIGH de media σ_2 .

Así, la relación SEÑAL/CLUTTER del blanco vendrá dada por la expresión:

$$S/C = \frac{\sigma_2^2}{\sigma_1^2} ;$$

Conociendo estas características podemos construir un modelo generador de números aleatorios con estadística tipo RAYLEIGH, de media la que el usuario decida caracterizar, mediante el METODO DE LA TRANSFORMACION [1] a partir del generador de números aleatorios con distribución uniforme que posee la biblioteca standard del lenguaje "TURBO C". Los números que se obtienen representarán la tensión de entrada a la etapa de F.I del RADAR, que se agruparán en archivos, cada uno identificado con un cuadrante, donde

cada muestra es de una longitud de 16 bits (un entero sin signo definido en TURBO C) , que facilita su posterior procesado.

Extendiendonos en el módulo DETECTOR por umbral fijo, éste se encargará de procesar dichos archivos mediante una sencilla comparación de la tensión simulada y asignada a cada célula de resolución con un nivel prefijado por el usuario y que nos determinará una probabilidad de falsa alarma y de detección fácilmente calculable numéricamente, y que estarán precalculados en memoria en forma de pares $P_{fa} \rightarrow T$, así se crearán los archivos asignados a cada cuadrante con información de tipo booleana, es decir "sí blanco" o "no blanco", cuyo resultado es almacenado en un byte de memoria.

Debido a los requerimientos de alta velocidad de salida y entrada de datos del PC hacia el RADAR, y viceversa, es preciso un lenguaje de programación para gobernar dichos procesos que sea de altas prestaciones, y qué mejor que el lenguaje "ASSEMBLER" que permite realizar todas las funciones anteriormente propuestas con eficiencia y velocidad, permitiendo además, tener un control directo sobre la velocidad de ejecución de un programa pues se conoce el "Timing" de cada instrucción en uso, así pues, los programas controladores de extracción y adquisición están desarrollados en este lenguaje.

A modo de descripción general, su desarrollo secuencial puede desglosarse en tres bucles, el primero de sincronización de paso por el ARP con el inicio de extracción de datos, necesario para el volcado sobre el primer cuadrante. El segundo es el de sincronización del barrido del P.P.I con la extracción de las muestras que la forman,. Y por último el volcado en sí de los datos junto con la activación del oscilador de gobierno de las básculas.

Esto programas en lenguaje "ASSEMBLER" son "linkados" con un programa en lenguaje de alto nivel "TURBO C" encargado de recoger los archivos de información que forman todo el espacio, cargarlos a memoria fija, con una dirección física conocida, y traspasar en los momentos oportunos la información de las direcciones a las subrutinas en "ASSEMBLER" anteriormente desarrolladas.

RESULTADOS:

Si bien el sistema presenta un comportamiento ajustado a los objetivos propuestos, es decir, se permiten visualizaciones del video crudo para luego observar las distintas pantallas procesadas, percibiéndose claramente los efectos positivos que comportan, sobretodo tanto el extractor como el C.F.A.R, se presentan los típicos problemas de usar un software soportado por un ordenador en configuración PC, una velocidad de procesado relativamente lenta, a modo de ejemplo la subrutina generadora de números aleatorios con estadística de Rayleigh crea una pantalla RADAR en aproximadamente 20 minutos (40 muestras por segundo), por ello la necesidad de poseer un banco de datos que correspondan a distintas pantallas. Si tratamos de la subrutina de carga de memoria, tenemos que el tiempo requerido para ese fin es de unos 8 segundos , por tanto, es imposible pensar en un simulador en tiempo real, pero dentro de lo que cabe el sistema tiene un comportamiento aceptablemente rápido.

BIBLIOGRAFIA

- [1] W. H. Press et al. *NUMERICAL RECIPES* CAMBRITGE UNIVERSITY PRESS
- [2] *MANUAL TECNICO DEL RADAR HALCON-948 DE HISPANO MARITIMA S.A*
- [3] *TURBO-C version 2.0 USER'S GUIDE Y REFERENCE GUIDE* BORLAND
- [4] *TURBO-ASSEMBLER version 1.0 REFERENCE GUIDE* BORLAND

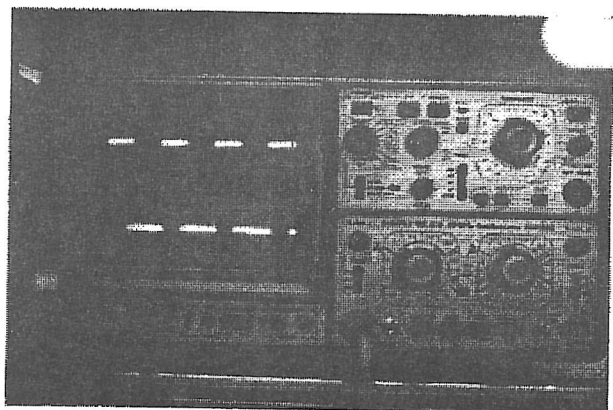


figura-1 señal cuadrada generada via software de 180Khz.

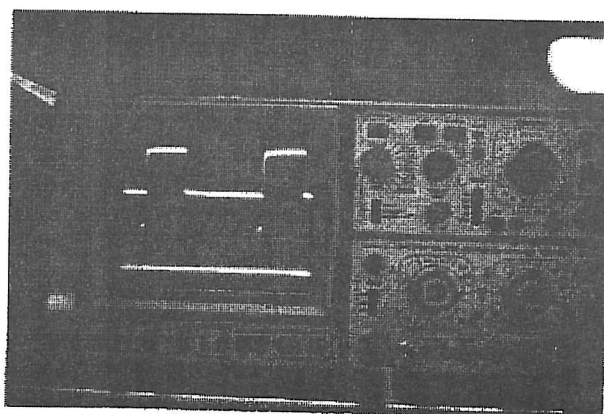


figura-2 detalle de la señal de barrido y sincronismo de lectura de línea.

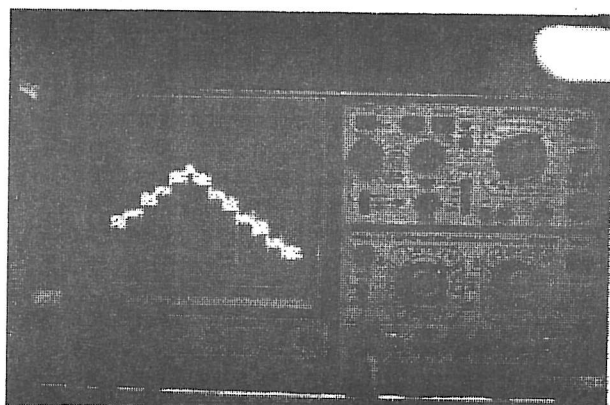


figura-3 señal triangular a la salida de D/A sin básculas.

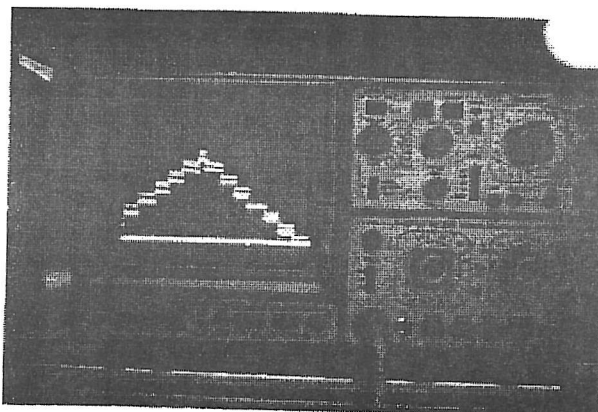


figura-4 señal triangular a la salida de D/A con básculas.

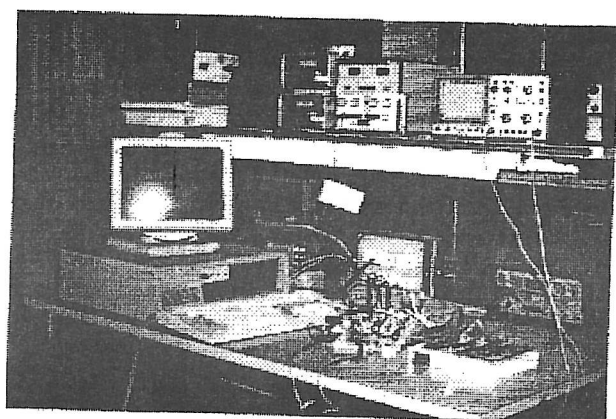


figura-5 detalle de puesto de trabajo, a la izquierda fuera de imagen se halla el P.P.I.